REDES NEURAIS COM TENSORFLOW

PERCEPTRON DE ROSENBLATT

DIEGO RODRIGUES DSC

INFNET

CRONOGRAMA

Dia	Aula	Trab
29/07	Perceptron de Rosenblatt	
31/07	Classificação: Neurônio Sigmóide	
05/08	Classificação: Rede Neural Feedforward	Grupos
07/08	Classificação: Treinamento Robusto	
12/08	Regressão	Base de Dados
14/08	Agrupamento	
19/08	Séries Temporais	Modelos
21/08	Apresentação dos Trabalhos Parte I	

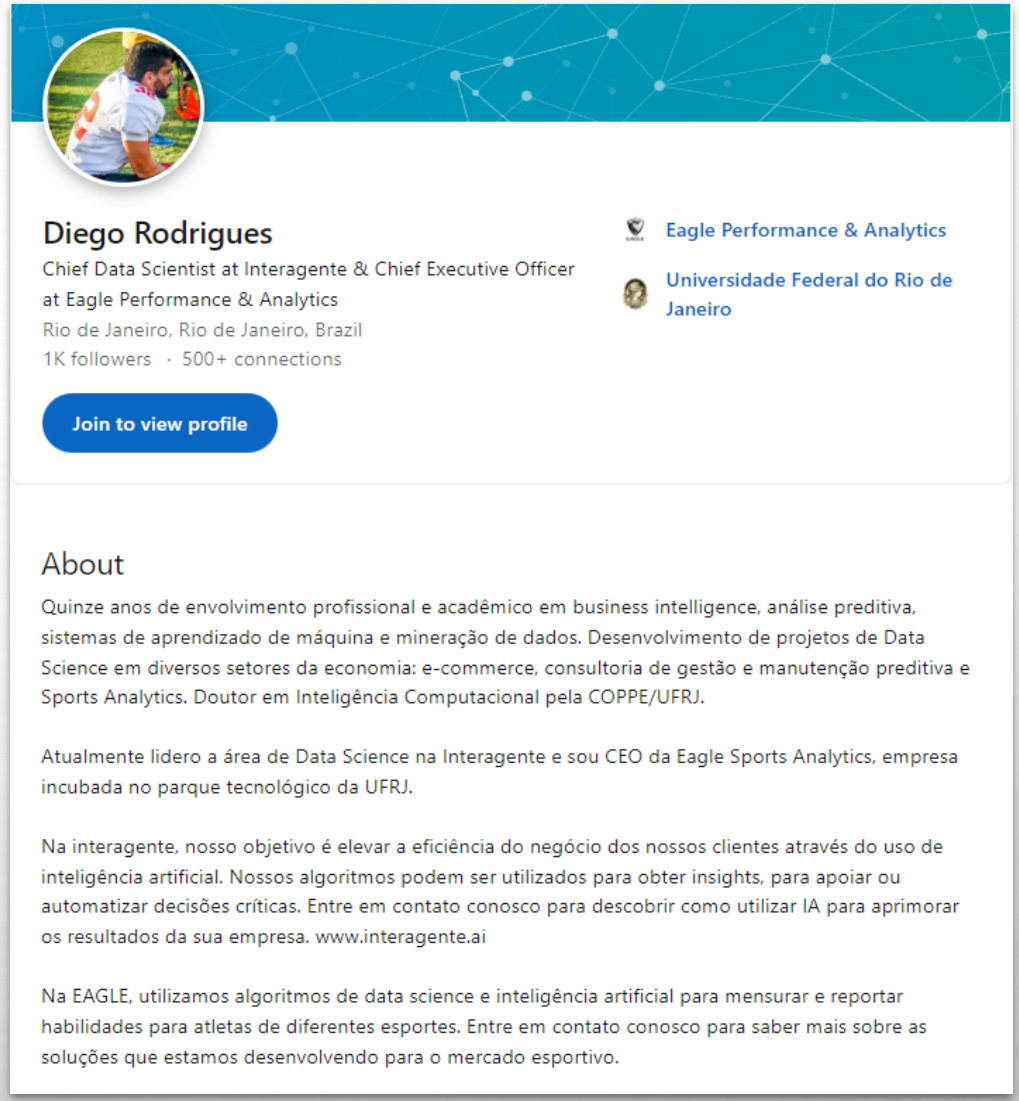
PERCEPTRON DE ROSENBLATT

- PARTE 1 : TEORIA
  - APRESENTAÇÃO DA TURMA
  - APRENDIZADO DE MÁQUINA
  - HISTÓRIA DAS REDES NEURAIS
  - O APROXIMADOR UNIVERSAL
  - FUNÇÕES DE ATIVAÇÃO
  - GOOGLE TENSOR FLOW PLAYGROUND
- PARTE 2 : PRÁTICA
  - AMBIENTE DE DESENVOLVIMENTO
  - BIBLIOTECAS
  - NOTEBOOK: PERCEPTRON
- PARTE 3 : TRABALHOS
  - ESCOPO & EVOLUÇÃO

PARTE 1 : TEORIA

PROFESSOR DIEGO

- Doutor em Inteligência Computacional COPPE / UFRJ
- 16 anos de experiência com ciência de dados aplicada à engenharia.
- Sócio da Interagente e Eagle Sports Analytics.
- Professor da INFNET na Graduação, Pós-Graduação, Trilhas e Bootcamp.



Diego Rodrigues

Chief Data Scientist at Interagente & Chief Executive Officer at Eagle Performance & Analytics

Rio de Janeiro, Rio de Janeiro, Brazil

1K followers · 500+ connections

Join to view profile

Eagle Performance & Analytics

Universidade Federal do Rio de Janeiro

About

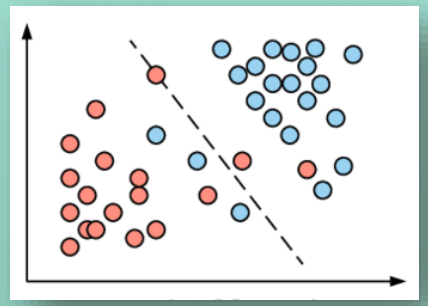
Quinze anos de envolvimento profissional e acadêmico em business intelligence, análise preditiva, sistemas de aprendizado de máquina e mineração de dados. Desenvolvimento de projetos de Data Science em diversos setores da economia: e-commerce, consultoria de gestão e manutenção preditiva e Sports Analytics. Doutor em Inteligência Computacional pela COPPE/UFRJ.

Atualmente lidero a área de Data Science na Interagente e sou CEO da Eagle Sports Analytics, empresa incubada no parque tecnológico da UFRJ.

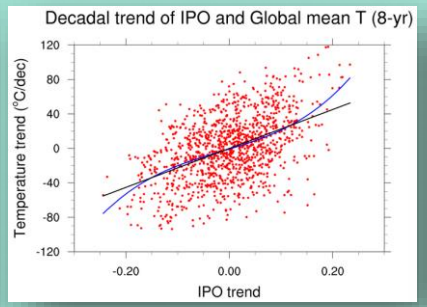
Na interagente, nosso objetivo é elevar a eficiência do negócio dos nossos clientes através do uso de inteligência artificial. Nossos algoritmos podem ser utilizados para obter insights, para apoiar ou automatizar decisões críticas. Entre em contato conosco para descobrir como utilizar IA para aprimorar os resultados da sua empresa. www.interagente.ai

Na EAGLE, utilizamos algoritmos de data science e inteligência artificial para mensurar e reportar habilidades para atletas de diferentes esportes. Entre em contato conosco para saber mais sobre as soluções que estamos desenvolvendo para o mercado esportivo.

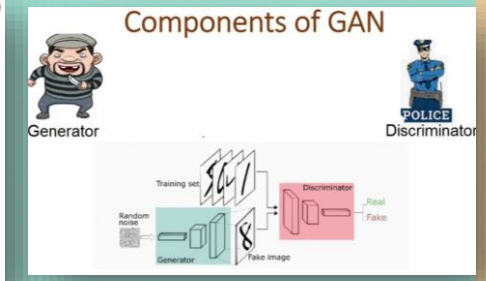
APRENDIZADO SUPERVISIONADO



CLASSIFICAÇÃO

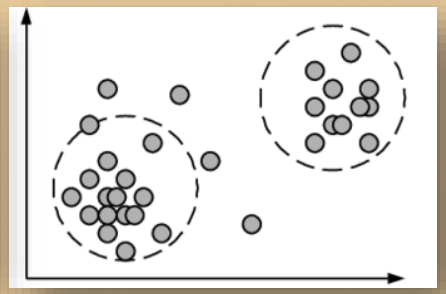


REGRESSÃO



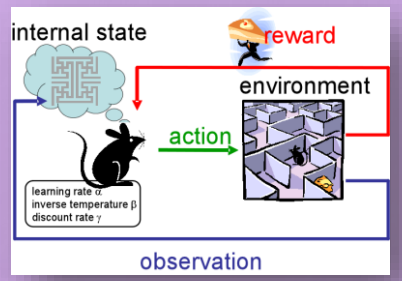
GENERATIVO

APRENDIZADO NÃO-SUPERVISIONADO



AGRUPAMENTO

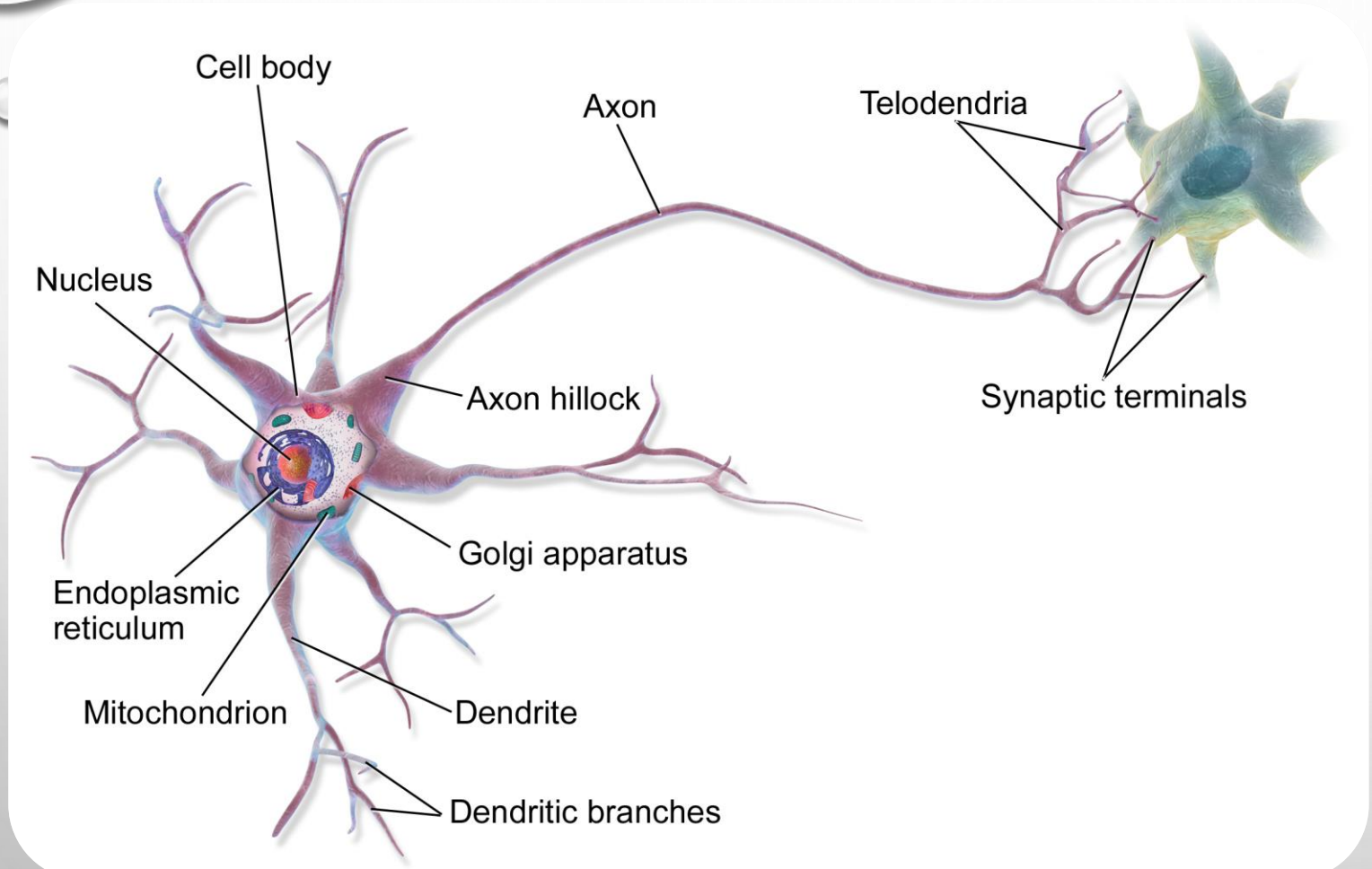
APRENDIZADO POR REFORÇO



REFORÇO

HISTÓRIA DAS REDES NEURAIS

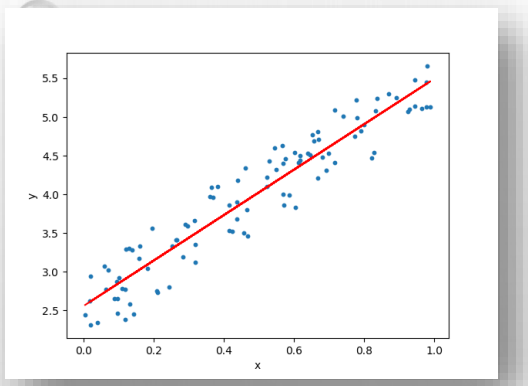
- Perceptron, Rosenblatt 1954
- Neocognitron, Fukushima 1979
- Backpropagation Rumelhart, Hinton & Williams 1986
- LSTM, Hochreiter e Schmidhuber1995
- Keras & Tensorflow, Google 2015



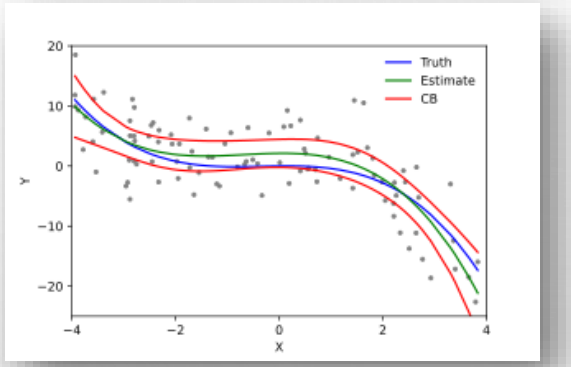
INSPIRAÇÃO BIOLÓGICA

O APROXIMADOR UNIVERSAL

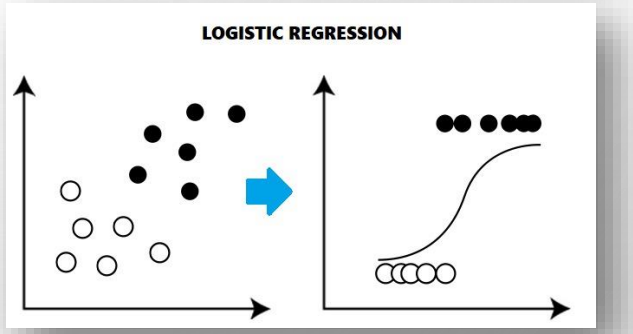
$$Y = \underbrace{F(X)}_{\text{Parte Determinística}} + \underbrace{\varepsilon}_{\text{Parte Estocástica}}$$



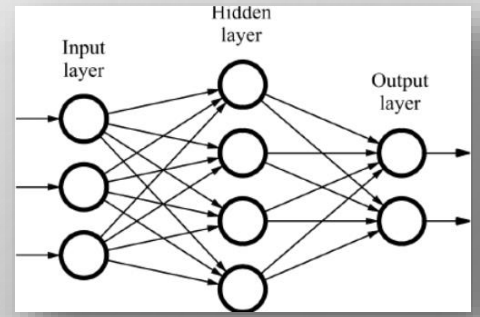
$$Y = \alpha^T x + \varepsilon$$



$$Y = X\alpha + \varepsilon$$

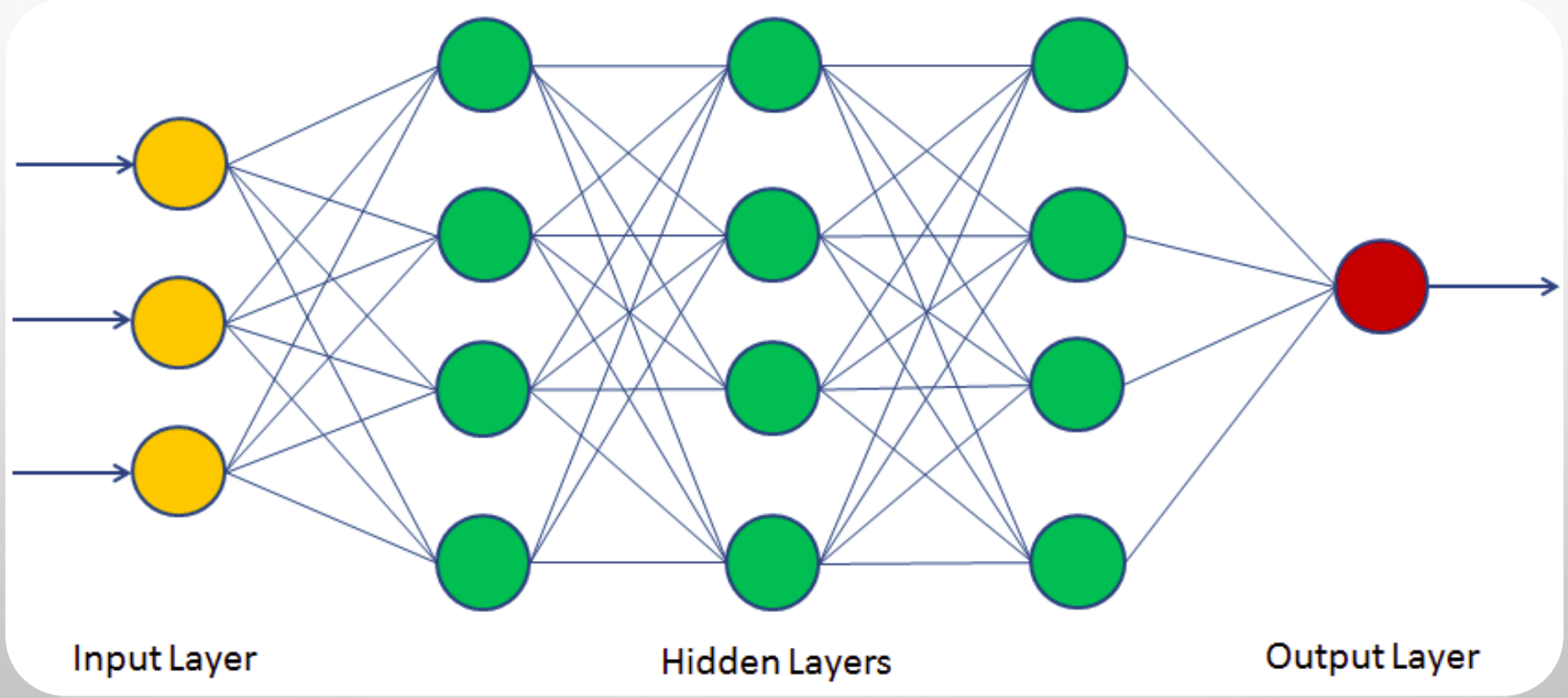


$$Y = \frac{1}{1 + e^{\alpha^t x + \varepsilon}}$$

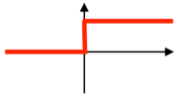
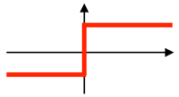
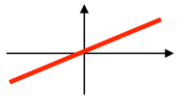
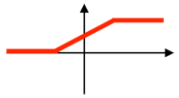
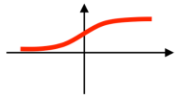
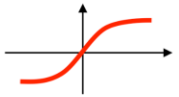
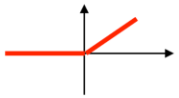
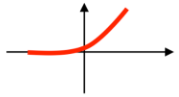


$$Y = \varphi(x) + \varepsilon$$

O APROXIMADOR UNIVERSAL



FUNÇÕES DE ATIVAÇÃO

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^{z})$	Multi-layer Neural Networks	

GOOGLE TENSORFLOW PLAYGROUND

PARTE 2 : PRÁTICA

AMBIENTE DE DESENVOLVIMENTO

Miniconda

Miniconda is a free minimal installer for conda. It is a small bootstrap version of Anaconda that includes only conda, Python, the packages they both depend on, and a small number of other useful packages (like pip, zlib, and a few others). If you need more packages, use the `conda install` command to install from thousands of packages available by default in Anaconda's public repo, or from other channels, like conda-forge or bioconda.

Is Miniconda the right conda install for you? The Anaconda or Miniconda page lists some reasons why you might want one installation over the other.

System requirements

Latest Miniconda installer links by Python version

Installing Miniconda

Miniconda release notes

Other resources

Miniconda hash information

Simple Python Version Management: pyenv

gitter join chat

pyenv lets you easily switch between multiple versions of Python. It's simple, unobtrusive, and follows the UNIX tradition of single-purpose tools that do one thing well.

This project was forked from rbenv and ruby-build, and modified for Python.

```
$ pyenv versions
  2.7.10
* 3.5.0 (set by /Users/yuu/.pyenv/version)
  miniconda3-3.16.0
  pypy-2.6.0

$ python --version
Python 3.5.0

$ pyenv global pypy-2.6.0

$ python --version
Python 2.7.9 (295ee98b69288471b0fcf2e0ede82ce5209eb90b, Jun 01 2015, 17:30:13)
[PyPy 2.6.0 with GCC 4.9.2]

$ cd /Volumes/treasuredata/jupyter

$ pyenv version
miniconda3-3.16.0 (set by /Volumes/treasuredata/.python-version)

$ python --version
Python 3.4.3 :: Continuum Analytics, Inc.
```

Boa Prática: 1 Ambiente por projeto!

CRIANDO AMBIENTE COM CONDA

Managing environments

With conda, you can create, export, list, remove, and update environments that have different versions of Python and/or packages installed in them. Switching or moving between environments is called activating the environment. You can also share an environment file.

There are many options available for the commands described on this page. For a detailed reference on all available commands, see commands.

Creating an environment with commands

Use the terminal for the following steps:

1. To create an environment:

```
conda create --name <my-env>
```

   Replace `<my-env>` with the name of your environment.

2. When conda asks you to proceed, type `y`:

```
proceed ([y]/n)?
```

   This creates the myenv environment in `/envs/`. No packages will be installed in this environment.

3. To create an environment with a specific version of Python:

```
conda create -n myenv python=3.9
```

Activating an environment

Activating environments is essential to making the software in the environments work well. Activation entails two primary functions: adding entries to PATH for the environment and running any activation scripts that the environment may contain. These activation scripts are how packages can set arbitrary environment variables that may be necessary for their operation. You can also use the config API to set environment variables.

Activation prepends to PATH. This only takes effect when you have the environment active so it is local to a terminal session, not global.

Note

When installing Anaconda, you have the option to "Add Anaconda to my PATH environment variable." *This is not recommended* because it *appends* Anaconda to PATH. When the installer appends to PATH, it does not call the activation scripts.

Note

On Windows, PATH is composed of two parts, the *system* PATH and the *user* PATH. The system PATH always comes first. When you install Anaconda for "Just Me", we add it to the *user* PATH. When you install for "All Users", we add it to the *system* PATH. In the former case, you can end up with system PATH values taking precedence over your entries. In the latter case, you do not. *We do not recommend* multi-user installs.

To activate an environment: `conda activate myenv`

https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

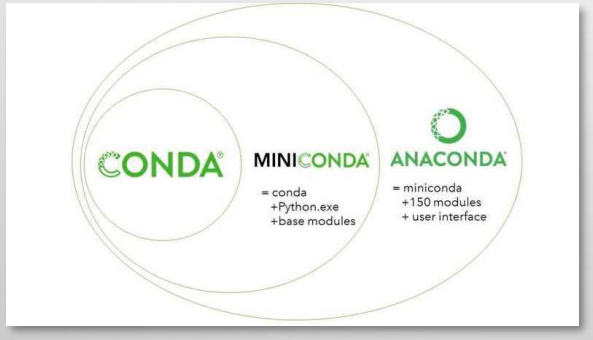
AMBIENTE PYTHON

5. Visualização

6. Machine Learning

4. Variáveis Aleatórias

1. Editor de Código

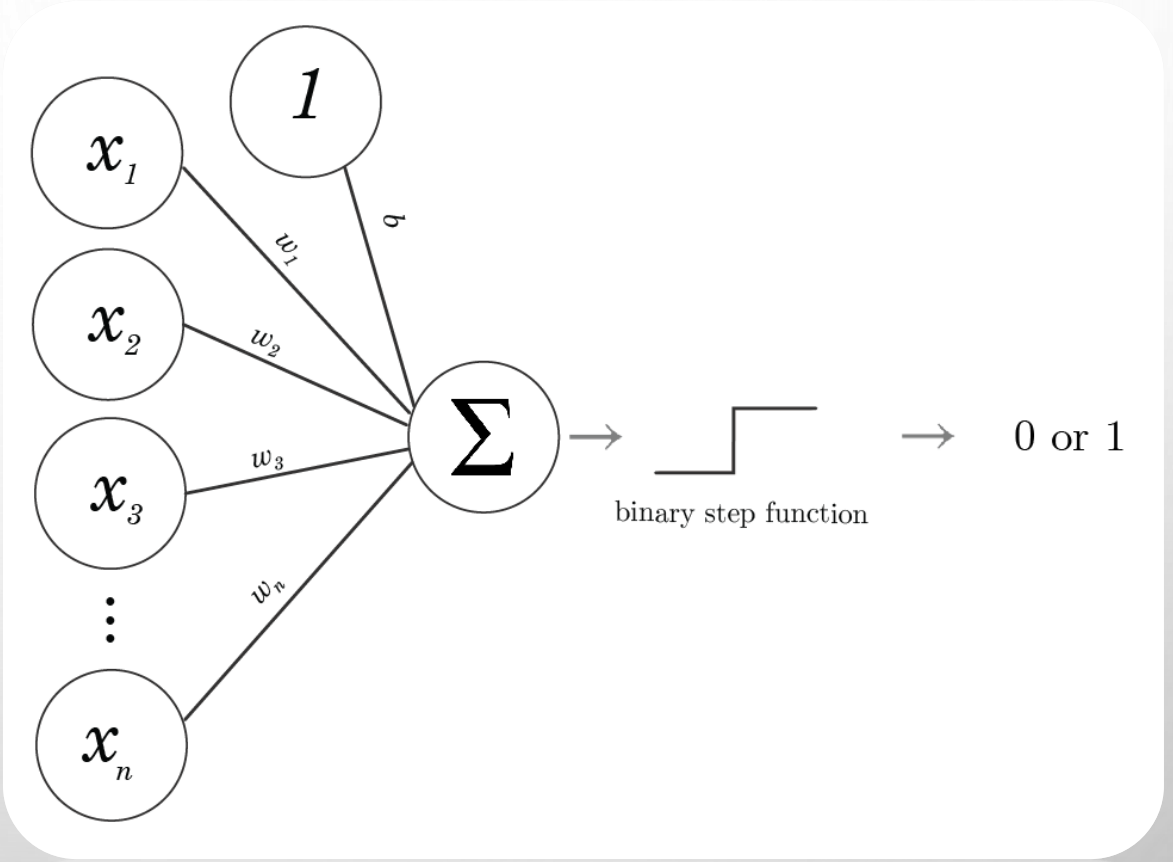


2. Gestor de Ambiente

3. Ambiente Python do Projeto

3. Notebook Dinâmico

PERCEPTRON DE ROSENBLATT



EXERCÍCIO: PERCEPTRON DE ROSENBLATT

PRÓXIMA AULA: CLASSIFICAÇÃO COM NEURÔNIO SIGMÓIDE